# Severity Assessment of Software Defect Reports using Text Classification

Ruchika Malhotra, Ph.D Assistant Professor Delhi Technological University Nidhi Kapoor B.Tech (Software Engg.) Delhi Technological University

# ABSTRACT

Defect severity assessment is essential in order to allocate testing resources and effectively plan testing activities. In this paper, we use text classification techniques to predict and assess the severity of defects. The results are based on defect description of issue requirements obtained from NASA project. We have used Support Vector Machine technique to predict defect severity from issue reports.

## **Keywords**

Text Classification, Severesis, PITS, SVM

## **1. INTRODUCTION**

Severity is defined as the impact of a failure on software. It reflects upon the degree to which the severity affects the functionality and the degree to which a defect can have impact on the development or operation of software. Therefore, the defect severity can be stated as the effect of a defect on a component or software. It underlines the criticality of the defect. The value of the severity varies in accordance with the intensity of the failure caused by the defect.

Defect severity assessment is mainly concerned with assessment of quality level of a software to see whether the software is fit enough to be released. Practically, the defect reports are filled manually on periodic basis. The value of severity levels assigned to defects manually could be different from the one computed by text mining and rule learning. To prevent a testing engineer from assigning wrong severity levels to a defect, to save time in case of an urgent-time bounded application, to check that no high severity problem has been missed, an automated extraction and analysis of text from issue reports has been employed [1].

In defect severity assessment, significantly important conclusions have been presented from the given NASA data set[4] using text mining and machine learning and an important problem is been addressed, which may not have been noticed by the research community. Unlike the extensive work already done in the same scenario where the data used was highly structured and indented, a computation technique to address and classify unstructured data has been presented in this paper. Initially the system is trained with two thirds of the data so as to make it understand the severity of the various defects encountered and then text classification technique is used to further predict the severity of defects of the rest of the data; thereby, implementing effective machine learning via Support Vector Machine.

## **2. RELATED WORK**

The purpose of assigning severity levels to defects is to govern the effect of that defect on the deployment of the software. As a matter of fact, different levels may be assigned to different projects and the developers of a product and the user may not agree on a particular level assignment. Such are the areas that need attention. Rishabh Jain B.Tech (Software Engg.) Delhi Technological University Sahaj Biyani B.Tech (Software Engg.) Delhi Technological University

An automatic approach that allows a quality analyst in assigning severity levels to defects is depicted in the paper by Menzies and Marcus and is named SEVERIS [2] (SEVERity ISsue a

ssessment). It involves application of text mining and machine learning techniques on a given set of defect reports. This approach involves reviewing of the issue reports automatically and alerting when a severity is found anomalous. The paper encapsulating the very approach highlights the study of the results encountered when SEVERIS is used with NASA's Project and Issue Tracking System (PITS) and thereby, reflects upon the fact that SEVERIS, being an easy-to-use and efficient approach, results in a considerably better prediction of defect severity levels. The idea behind the research in this paper is same as that of Menzies and Marcus's; but, unlike them, the Support Vector Machine has been used to implement machine learning algorithm along with some changes in the feature selection technique.

Zhou and Leung [4] too have carried out a substantial amount of work in the field of defect severity assessment. In order to examine the accuracy of the defect prediction, taking defect severity into account, they assessed the precision of defectproneness prediction for object oriented design metrics using logistic regression and machine learning methods. Their results indicate that there exists a statistical relation between design metrics and fault-proneness of classes for different severity levels.

Another interesting research related to this work comes from the research conducted by Suffian who used Six sigma methodology to maintain a defect-prediction model for the testing phase of software design. This research was carried out with the motto of zero defects to be encountered after the software is released to the end user. This motto was achieved by requirement elicitation from customers in order to draft the prediction model and by underlining the various factors that could assist in or associate with the discovery of defects in the testing phase. The extensive work carried out in the research helps in predicting the total number of defects irrespective of the severity of the defects and the time given to each activity in the testing phase. In the approach represented in this paper, the Suffian's approach has been extended for the prediction of the severity level not only at the testing phase but also at the system level testing phase and maintenance phase.

Additional motivation for the work comes from another research conducted to predict defects in the design phase itself. It uses the tool MetricView which visualizes the software metrics that have been created by an external tool and presents perception of UML models; thereby, making it easy to see what is present in the model and telling about what is not present. It can be considered similar to that which is mentioned in this paper as this too involves the use of software process though not for prediction of defects but for prediction of severity levels of defects i.e. the impact of the defects.

# 3. RESEARCH BACKGROUND

## **3.1 Proposed Technique**

As the document is first sent into the input, it is read and sent for being pre-processed which involves the reduction of attribute so that the irrelevant and unnecessary words present in the large amount of data could be limited to only the attributes that are significantly important. As depicted in figure 1, the preprocesed words are sent for feature selection so as to maintain a level of consistency among the attributes by ranking them from most important to significantly less important. The attributes are then applied with machine learning algorithm so as to classify the attributes on the basis of severity levels of their defects. The results obtained from machine learning are then evaluated using f-measure, which encompasses precision and recall, to know the degree to which the results obtained adhere to the desired output.



Figure 1. : Text Classification Process

Reducing the number of attributes so as to get only the relevant and significantly important attributes in order to save time in processing the entire chunk of data and shift our focus on relevant attributes. Data pre-processing encompasses three steps in order as: tokenization, stop word removal, and stemming[4]



Figure 2. Steps of Pre-Processing

## A. Tokenization

It is the task of converting a stream of characters into a stream of processing units called tokens, which are the blocks of text considered useful part of the unstructured text. In severity defect assessment tool, the main focus of tokenization is to identify the meaningful keywords by replacing punctuations, brackets, hyphen, capitals etc with blank spaces, removing non-printable escape characters, and converting all words to lower case. Besides this, tokenization can ensure that the document is consistent [2].

## B. Stop Word Removal

Stop words are the most common words that are not likely to carry any significant information in a particular context and are, therefore, not necessary for text mining application and are eliminated. The process of stop word removal reduces the text data and improves the performance of the system.

Some stop words in the English dictionary are: 'a', 'an', 'the', 'is', 'I', 'you', 'of', etc. These words are clearly of no use in our task of text classification. In defect severity assessment, we use a maintained list of stop words to eliminate them when encountered in the **data set**.

#### C. Stemming

Stemming or lemmatization is the process for reduction of words (or derived words) into their root form or stem regardless of their part of speech or the context of occurrence. For instance, agreed, agreeing, disagree, disagreement all belong to the stem agree.

There are two errors possible in stemming- over stemming and under stemming. Over stemming, also called false positive, occurs when two words with different stems stem to the same root. For example, despite being different in context of usage, both news and new return new after stemming [5]. Understemming, also known as false negative, occurs when two words that should have stemmed to the same root, have not done so.

The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root [2].

## **D.** Feature Selection

Even after data pre-processing on a document, the number of attributes is so large that it further needs to be reduced. Feature selection encapsulates the technique for finding the importance of a word on the basis of various techniques such as InfoGain. In this research, the technique of information gain has been used to do so. The aim is to minimize the entropy of the system and maximize the gain to rank the attributes. Next, the weights are assigned to the ranked attributes so as to compute the frequency of an attribute in a document with respect to the percentage of the frequency in the entire data set. This is done in order to get the weighted ranked attributes. The following figure shows the steps of feature selection as explained below.



## Figure 3: Steps of Feature Selection

## E. Applications in Machine Learning

Machine learning accelerates the task of text classification. The algorithm requiring probabilistic approach had been used extensively but mainly result in poor performance of the component or software involved[3]. In this research, we have applied Support Vector Machine algorithm of machine learning to classify documents on the basis of severity levels

It is suited for categorizing a considerably large amount of complex data. In our research, we have taken the advantage of the principle of supervised learning as incorporated in SVM. We firstly train the system with two-thirds of the data set and then classify unseen documents on the basis of the observation of classification of trained set of documents.

#### F. Evaluation of Machine learning results

Machine learning algorithms predict the classes of documents on the basis of the documents fed in the training phase. It is not necessary that it would always result in the correct classification of documents. The degree to correctness of the predicted classes is termed as predictive accuracy. It has now become an unconventional medium to compute the accuracy of a classifier since it is not a reliable indicator of the effectiveness of the classifier.

In our paper, we have explored the f-measure technique of accuracy computation which consists of computation of recall and precision. Recall underlines the proportion of documents in the given set of documents that have been rightly predicted. Precision as defined by Max Bramer in his book Principles of Data Mining, Springer is given as "the number of documents predicted for a level and is actually belonging to that level of severity".

Instead of computing merely predictive accuracy, the two approaches of Precision and Recall have been combined to correctly evaluate the result of machine learning.

#### G. Software Defects

As defined by Society, 1990, a defect is defined as "an incorrect step, process or data definition in a computer program." Software defects may incur during any phase of software development life cycle. These defects are called latent defects and are very hard to assess.

## **3.3 Defect Tracking System**

It is an essential step for project management. A defect report manifests the way a user sees the defect. Throughout the life cycle of the defect report, it goes through a workflow which consists of six stages as: NEW, EXAMINED, CONFIRMED, RESOLVED, VERIFIED and ACCEPTED.

When a defect report is is submitted into the system, it is first categorized as NEW. It is then assured for quality management by the quality assurance personnel, then it is sent into the EXAMINED state. Now, when the defect report is found to be causing a failure in the operation of the component or a system, it is sent to the CONFIRMED state. If a solution is determined to address the defect encountered, the defect report is moved into the RESOLVED status. The quality assurance personnel then verifies the solution to correct the defect and if the correction is done correctly, the defect report now moves to the VERIFIED state. Now the solution is presented to the client to be approved; if it is approved, the defect is transferred to the ACCEPTED state.

There can be a case when a defect report contains ways to correct the defect, it can directly be transferred from NEW or EXAMINED to RESOLVED. Even after a defect is in CONFIRMED state and the information present has been found to be incorrect, it can trace back to EXAMINED state for further processing from CONFIRMED. If the correction of the defect is not correct, it is reopened from RESOLVED state to EXAMINED. [2]

## 4. RESULT

The PITS A dataset was used for coming out with the result. 70% of the documents were used for training and the rest 30% were used for testing the accuracy and efficiency of the training method and thus predicting the effectiveness of the text classification algorithm.

The Support Vector Machine learning algorithm was implemented using the LIBSVM tool for multiclass classification. The results for a feature set of 5, 25, 50, 100 words were obtained and also was their corresponding accuracy. An increasing trend in the accuracy of the prediction was seen as the Features were increased.

Rank	Feature	
1	rvm	
2	question	
3	lead	
4	script	
5	whether	
6	baseline	
7	engcntrl	
8	sr	
9	test	
10	3	
11	operation	
12	set	
13	sfs	
14	reference	
15	с	
16	requirement	
17	14	
18	file	
19	dh	
20	miss	
21	line	
22	ignore	
23	approve	
24	link	
25	srobc	

Top 25 terms in data set after Feature Selection

#### Total 5 Features Selected

When selected total 5 features, we got an accuracy of 20 percent. The below figure is a plot of the training data based on SVM using LIBSVM.



Plot 1. Training for 5 Features

#### Total 50 Features Selected

When selected total 50 features, we got an accuracy of 48 percent. The below figure is a plot of the training data based on SVM using LIBSVM.



Plot 2. Training for 50 Features

#### Total 100 Features Selected

When selected total 100 features, we got an accuracy of 84 percent. The below figure is a plot of the training data based on SVM using LIBSVM.



Plot 3. Training for 100 Features

**RESULT BASED ON NUMBER OF FEATURES** 

No. of Features	Documents		
	Accuracy Percentage	Correct	Total
5	20	51	250
25	34	83	240
50	48	120	250
100	84	169	200

## **5. CONCLUSIONS AND FUTURE WORK**

In this research, text classification for assessing the severity levels of the defect reports and predicting the severity levels of unseen defect reports in real time has been used. In future this research is intended to use this tool for non-functional requirements report and maintenance request report.

## **6. REFERENCES**

- [1] Martin Pavlov ILIEV, "A method for automated prediction of defect severity using ontologies" in Master's thesis, LIACS, Leiden University, Logica Netherlands, 2012
- [2] Tim Menzies and Andrian Marcus, "Automated severity assessment of software defect reports", 2007
- [3] Konstantin Mertsalov and Michael McCreary, "Document classification with support vector machines", January 2009
- [4] Principles of data mining by Max Barmer, Springer
- [5] Yogesh Singh, Arvinder Kaur, Ruchika Malhotra ,"Software Fault Proneness Prediction Using Support Vector Machines"
- [6] Y.H Li and A.K Jain, "Classification of text documents",1998
- [7] Thorsten Joachims, "Text categorization with Support Vector Machines: learning with relevant features"
- [8] Shrikanth Shankar and George Karypis, "A feature weight adjustment algorithm for document categorization"